Tampereen yliopisto

Miika Luiro

PILVIPALVELUPOHJAISTEN TEKSTIN-TUNNISTUSJÄRJESTELMIEN SOVEL-TUVUUS TOSITTEIDEN KÄSITTELYYN

Tieto- ja sähkötekniikan tiedekunta
Kandidaatintyö
Kesäkuu 2019

TIIVISTELMÄ

Miika Luiro: Pilvipalvelupohjaisten tekstintunnistusjärjestelmien soveltuvuus tositteiden käsittelyyn, Suitability of cloud-based text recognition systems for processing receipts
Kandidaatintyö
Tampereen yliopisto
Tieto- ja sähkötekniikan kandidaatin tutkinto-ohjelma
Kesäkuu 2019

Tässä työssä tutkittiin, kuinka hyvin pilvipalvelupohjaiset tekstintunnistusjärjestelmät soveltuvat mobiililaitteilla kuvattujen tositteiden käsittelyyn. Työssä vertailtiin kahden eri palveluntarjoajan tekstintunnistusjärjestelmiä, joita olivat Google Cloud Vision ja Microsoft Azure Computer Vision. Vertailussa käytettävät tositteet valittiin eTasku Solutions Oy:n ylläpitämästä tositearkistosta.

Soveltuvuusvertailun lisäksi työssä esiteltiin tekstintunnistusprosessin keskeisimmät vaiheet, joita ovat kuvien hankinta, esikäsittely, segmentointi, piirreirrotus, luokittelu sekä jälkiprosessointi. Lisäksi käytiin läpi tekstintunnistuksen historiaa ja esiteltiin merkittävimpiä tekstintunnistusjärjestelmiä eri vuosikymmeniltä.

Soveltuvuusvertailun tuloksien perusteella selvisi, että mobiililaitteilla kuvattujen tositteiden käsittely pilvipalvelupohjaisilla tekstintunnistusjärjestelmillä on mahdollista. Erityisesti Google Cloud Vision -palvelun tulokset olivat lupaavia. Soveltuvuusvertailun avulla löydettiin myös muutamia tositteita, joiden tunnistamisessa palveluilla oli erityisesti ongelmia.

Avainsanat: tekstintunnistus, pilvipalvelu, tosite, kuitti

SISÄLLYSLUETTELO

1. JOHDANTO

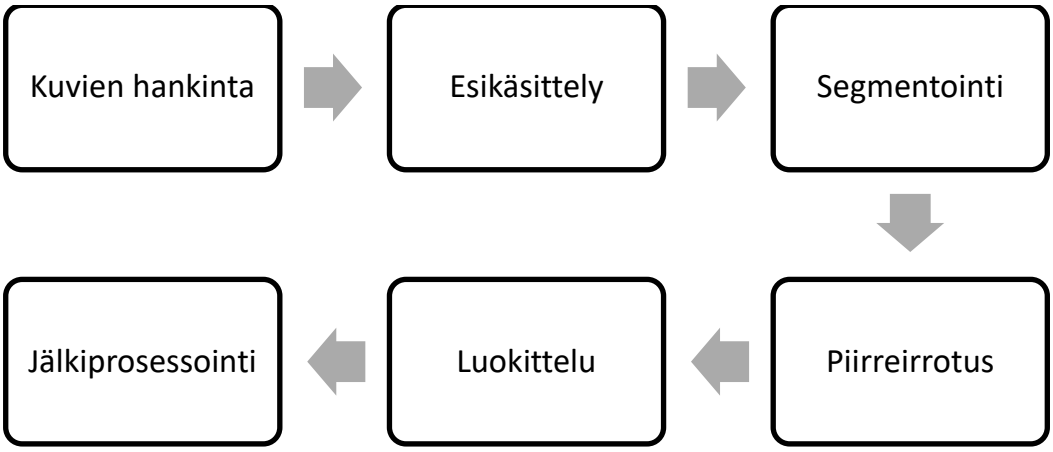
Tekstintunnistusta on hyödynnetty pitkään muun muassa kaavakkeiden ja passien käsittelyssä. Tämän tyyppisten dokumenttien käsittely on ollut mahdollista, koska niiden rakenne on ennalta tarkkaan määritelty. Huonolaatuisten ja rakenteeltaan vaihtelevien dokumenttien käsittely tekstintunnistuksen avulla on ollut haastavampaa. Tähän on kuitenkin tullut muutos 2010-luvun aikana. Useat yritykset ovat viime vuosina julkaisseet pilvipalveluympäristöissä toimivia tekstintunnistusjärjestelmiä. Nämä palvelut mahdollistavat tekstintunnistuksen laitteissa, joissa itsessään sen toteuttaminen olisi haastavaa kuten mobiililaitteissa.

Tämän työn tarkoituksena on vertailla yleisimpien pilvipalvelupohjaisten tekstintunnistusjärjestelmien soveltuvuutta tositteiden käsittelyyn. Vertailussa käytetään mobiililaitteilla kuvattuja tositteita. Työn keskeisin tavoite on selvittää, millä todennäköisyydellä tositteista saadaan tekstintunnistuksen avulla tarvittavat tiedot. Samalla voidaan vertailla palveluiden keskinäisiä tarkkuuksia. Vertailujen perusteella tiedetään, soveltuuko pilvipalvelupohjainen tekstintunnistus yleisesti eri laatuisten tositteiden käsittelyyn ja mikäli soveltuu, voidaan valita tähän käyttötarkoitukseen parhaiten sopiva palvelu.

Luvussa 2 käydään läpi, miten tekstintunnistusjärjestelmät käytännössä toimivat ja kuinka ne ovat historian aikana kehittyneet. Luvussa 3 esitellään kaksi pilvipalvelupohjaista tekstintunnistusjärjestelmää, joita vertaillaan tämän työn soveltuvuusvertailussa. Luvussa 4 käydään läpi soveltuvuusvertailun suorittamiseen liittyvät asiat kuten aineiston ja tärkeimpien vertailukriteerien esittely sekä analysoidaan vertailusta saatuja tuloksia. Luvussa 5 tiivistetään soveltuvuusvertailusta muodostetut johtopäätelmät ja esitellään keinoja, joilla soveltuvuusvertailua voisi tulevaisuudessa kehittää.

2. TEKSTINTUNNISTUS (OCR)

Tekstintunnistus on hahmontunnistuksen osa-alue. Tekstintunnistuksen tarkoituksena on muuttaa erityyppisten dokumenttien sisältämät tekstit tietokoneelle ymmärrettävään muotoon. Prosessoitavat dokumentit voivat olla sekä käsin kirjoitettuja että tietokoneella tulostettuja [1]. Tässä työssä keskitytään erityisesti tulostetuista dokumenteista otettujen kuvien prosessointiin.



Kuva 1. *Tekstintunnistusprosessin päävaiheet, muokattu lähteistä [1,2].*

Tekstintunnistus koostuu useasta eri vaiheesta, jotka ovat esitelty kuvassa 1. Prosessoitavan dokumentin sisältämien merkkien tunnistus tapahtuu luokitteluvaiheessa, mutta tätä ennen suoritetaan vaiheita, jotka muun muassa parantavat tekstintunnistuksen tarkkuutta ja poistavat luokittelun kannalta epäoleellisen informaation. [22]

Tekstintunnistusjärjestelmien tarkkuuden arviointiin käytetään usein merkkisuhdetta (CER) ja sanavirhesuhdetta (WER). Virhesuhteet kertovat, kuinka monta lisäystä, poistoa tai muutosta saatuun tekstiin on tehtävä suhteessa tekstin pituuteen, jotta se vastaa alkuperäistä tekstiä. Merkkivirhesuhde voidaan esittää kaavalla

$$CER = \frac{I+D+S}{N} * 100, \qquad (1)$$

jossa I on pienin tarvittava määrä lisättäviä merkkejä, D on pienin tarvittava määrä poistettavia merkkejä, S on pienin tarvittava määrä muutettavia merkkejä ja N on merkkien kokonaismäärä. [5]

Sanavirhesuhde esitetään saman tyyppisellä kaavalla kuin merkkivirhesuhde. Sanavirhesuhde voidaan esittää kaavalla

$$WER = \frac{I+D+S}{N} * 100, \quad (2)$$

jossa I on pienin tarvittava määrä lisättäviä sanoja, D on pienin tarvittava määrä poistettavia sanoja, S on pienin tarvittava määrä muutettavia sanoja ja N on sanojen kokonaismäärä. [5]

2.1 Kuvien hankinta ja esikäsittely

Tekstintunnistusprosessi alkaa kuvien hankinnalla. Prosessoitavat dokumentit täytyy ensin muuttaa digitaaliseen muotoon. Tähän käytetään yleensä joko skanneria tai kameraa. [3]

Useimmat dokumentit sisältävät luonnostaan vain kahta eri väriä. Tästä syystä kuvien binarisointi (engl. binarization) eli mustavalkoiseksi muuttaminen on perusteltua. Binarisoinnin avulla saadaan huomattavasti vähennettyä kuvan sisältämää ylimääräistä informaatiota. Tämä yksinkertaistaa seuraavissa vaiheissa suoritettavia dokumentin segmentointia ja merkkien tunnistusta. [3, 4]

Prosessoitavien dokumenttien sisältämä teksti on usein vinossa. Jotta dokumentin sisältämät tekstirivit voitaisiin tunnistaa oikein, täytyy dokumentti ensin suoristaa vinouman tunnistuksen ja poiston (engl. skew detection and correction) avulla. [14] Vinossa olevat tekstirivit heikentävät tekstintunnistuksen tarkkuutta huomattavasti. Jirasuwankul esittää tutkimuksessaan, että Arial-fontilla kirjoitetuilla dokumenteilla 5°:n vinouma pudottaa tarkkuuden 92 %:iin alkuperäisestä ja 10°:n vinouma 15 %:iin [15].

2.2 Segmentointi

Segmentoinnissa prosessoitava dokumentti jaetaan homogeenisiin osiin, jotka sisältävät vain tietyntyppistä informaatiota [17]. Tärkeimpiä dokumentista tunnistettavia osia ovat muun muassa otsikot, sarakkeet, taulukot ja tekstirivit. Segmentoinnin seurauksena dokumentista muodostuu looginen puurakenne [16]. Segmentoinnissa käytetyt algoritmit voidaan jakaa kolmeen eri luokkaan: ylhäältä alaspäin etenevät algoritmit (engl. top-down approach), alhaalta ylöspäin etenevät algoritmit (engl. bottom-up approach) ja hybridialgoritmit (engl. hybrid approach). Hybridialgoritmi on yhdistelmä ylhäältä alaspäin ja alhaalta ylöspäin etenevistä algoritmeista. [17]

Ylhäältä alaspäin etenevä algoritmi suorittaa segmentoinnin jakamalla dokumentin rekursiivisesti pienempiin osiin, kunnes ennalta määriteltyjen kriteerien perusteella saavutetaan pienin mahdollinen kokonaisuus, jota ei voi enää jakaa pienemmäksi [17]. Nagy et al. [19] vuonna 1992 esittelemä rekursiivinen X–Y-leikkausalgoritmi (RXYC) on ylhäältä alaspäin etenevä algoritmi, joka muodostaa tutkittavasta suorakulmaisen muotoisesta alueesta projektiot x- ja y-suunnassa. Projektioissa valkoiset pikselit kuvataan nollilla ja mustat pikselit ykkösillä. Tutkittava alue jaetaan pienempiin osiin kohdista, joissa valkoiset pikselit muodostavat ennalta määritettyä raja-arvoa pidemmän ketjun. Tätä jatketaan, kunnes koko dokumentti on jaettu mahdollisimman pieniin suorakulmion muotoisiin osiin. [18, 19] Shafait et al. suosittelevat tutkimuksessaan [18] käyttämään RXYC-algoritmia hyvälaatuisiin dokumentteihin, jotka eivät sisällä suuria vinoumia.

Alhaalta ylöspäin etenevä algoritmi aloittaa segmentoinnin prosessoitavan kuvan pikseleistä ja kokoaa näistä isompia kokonaisuuksia [17]. Shafait et al. esittelevät tutkimuksessaan [18] muutamia tämän tyyppisiä algoritmeja: Voronoi-diagrammiin perustuva algoritmi ja Docstrum-algoritmi. Molemmat algoritmit sopivat RXYC-algoritmia paremmin heikkolaatuisten dokumenttien segmentointiin. Voronoi-diagrammiin perustuva algoritmi ja Docstrum-algoritmi eivät suoriudu hyvin erikokoisia ja -tyylisiä fontteja sisältävien dokumenttien segmentoinnista. Tämän tyyppisten dokumenttien segmentointiin Shafait et al. suosittelevat rajoitettua tekstirivin tunnistusalgoritmia (engl. constrained text-line detection algorithm). [18]

2.3 Piirreirrotus

Piirreirrotuksessa segmentoinnin tuloksena saaduista merkeistä muodostetaan piirrevektoreita (engl. feature vector), jotka yksilöivät dokumentin sisältämät eri merkit. Piirreirrotukseen käytettävät menetelmät on jaettu kolmeen eri pääkategoriaan: tilastolliset piirteet (engl. statistical features), globaalit muunnos- ja sarjakehittämispiirteet (engl. global transformation and series expansion features) sekä geometriset ja topologiset piirteet [20].

Tilastollisia piirteitä voidaan etsiä muun muassa jakamalla tutkittava merkki pienempiin alueisiin (engl. zoning). Tämän jälkeen voidaan analysoida esimerkiksi kaarevien kohtien ja risteyskohtien esiintymistiheyttä eri alueilla. [20, 21] Tutkittavasta merkistä voidaan myös muodostaa projektiovektoreita projisoimalla merkin pikseleiden harmaa-arvoja eri suuntaisille suorille [21].

Fourierin muunnos (engl. Fourier transform) on yleisesti käytetty tekniikka globaalien muunnos- ja sarjakehittämispiirteiden erotteluun. Fourierin muunnoksen avulla merkit

voidaan esittää taajuustasossa. Matala taajuustaso sisältää tietoa merkin yleisistä ominaisuuksista ja korkeammalta taajuustasolta saadaan tietoa tarkemmista yksityiskohdista. Toinen esimerkki globaaleista muunnos- ja sarjakehittämispiirteistä on Hough muunnos (engl. Hough transform). Hough muunnoksen avulla voidaan tunnistaa tekstirivien perustaso (engl. baseline) ja analysoida yksittäisten merkkien kaarevuutta. [20]

Kumar et al. mukaan latinalaisessa merkistössä merkit koostuvat kuvassa 2 esitetyistä viivoista. Geometrisiä ja topologisia piirteitä voidaan muodostaa erottelemalla tutkittavista merkeistä näitä viivoja [20].

Kuva 2. *Viivat, joita käytetään geometristen piirteiden tunnistuksessa [20].*

2.4 Luokittelu

Luokittelu on tekstintunnistuksen keskeisin osa. Luokittelun tarkoituksena on yhdistää piirreirrotuksen avulla muodostetut piirrevektorit ennalta määriteltyihin luokkiin [22]. Tekstintunnistuksen tapauksessa nämä luokat voivat olla esimerkiksi latinalaisen merkistön eri merkkejä. Merkkien luokittelusta vastaavat eri tyyppiset luokittelijat. Mahdollisia luokittelijamenetelmiä ovat muun muassa tilastolliset menetelmät, neuroverkot (engl. neural networks) ja tukivektorikone (engl. support vector machine) [22].

Eräs tilastollinen luokittelumenetelmä on k-NN-sääntö. Kyseisessä menetelmässä tutkittavaa piirrevektoria verrataan harjoitusvaiheessa luokiteltuihin piirrevektoreihin. Tutkittavan piirrevektorin luokaksi valitaan lähimpänä olevan ennalta luokitellun piirrevektorin luokka. Toinen yleisesti käytetty tilastollinen luokittelija on Bayesilainen luokittelija. [30]

Yleisiä neuroverkkoihin pohjautuvia luokittelijoita ovat muun muassa eteenpäin syöttävät (engl. feed-forward network) ja radiaalipohjaiset funktio (engl. radial-basis function) neuroverkot [30]. Yetirajam et al. tutkivat eteenpäin syöttävien neuroverkkojen hyödyntämistä rikkinäisten merkkien tunnistamisessa [31]. Tutkimuksen mukaan tämän tyyppisellä neuroverkolla rikkinäisten merkkien tunnistustarkkuus nousi 45 %:sta 68 %:iin. Yetirajam et al. mukaan eteenpäin syöttävät neuroverkot ovat käytännöllisiä, koska niiden harjoittaminen onnistuu pienemmällä datamäärällä kuin monimutkaisempien neuroverkkojen.

Yksittäisen luokittelijan kehittäminen optimaaliselle tasolle vaatii huomattavan suuren määrän harjoitusdataa. Tämän takia on käytännöllisempää käyttää useampaa luokittelijaa. Lopullinen luokittelutulos muodostetaan yhdistämällä näiden eri luokittelijoiden tulokset. [22]

2.5 Jälkiprosessointi

Luokittelija ei välttämättä onnistu erottelemaan kirjaimia tekstistä täydellisesti. Varsinkin monimutkaisten kielten kohdalla virheellisiä tulkintoja syntyy enemmän [2]. Esimerkiksi Googlen Cloud Vision -palvelussa japanin kielisten dokumenttien normalisoitu merkkivirhesuhde (N-CER) on 4,9 %, kun taas englannin kielisten dokumenttien vastaava luku on vain 0,6 %. Normalisoidussa merkkivirhesuhteessa ei oteta huomioon muun muassa pilkuista ja lainausmerkeistä johtuvia virheitä. [12]

Tekstintunnistuksen tarkkuutta pyritään jälkiprosessoinnissa parantamaan muun muassa oikeinkirjoitukseen ja sanakirjoihin pohjautuvilla todennäköisyysmalleilla [2]. Useamman vuosikymmenen ajan käytössä ollut N-gram todennäköisyysmalli pyrkii ennustamaan seuraavaa sanaa pohjatuen edellisiin sanoihin [6]. Viimeisimmän vuosikymmenen aikana kehittyneet syvät neuroverkot (engl. deep neural networks) ovat mahdollistaneet uusien jälkiprosessointitekniikoiden syntymisen. Mokhtar et al. käsittelevät tutkimuksessaan neuroverkkoihin pohjautuvan konekääntämisen (engl. neural machine translation) hyödyntämistä tekstintunnistuksen virhesuhteen pienentämisessä. Tutkimuksen mukaan tehokkaimmaksi malliksi osoittautui normalisoitu merkkipohjainen malli (engl. character based model with normalization). Tämä malli pienensi englannin kielisten dokumenttien merkkivirhesuhdetta (CER) 2,71 % ja sanavirhesuhdetta (WER) 14,88 %. [5]

2.6 Historia

Tekstintunnistuksen historia ulottuu 1900-luvun alkupuolelle. Fournier d'Alben vuonna 1913 kehittämä Optophone oli ensimmäinen laite, joka pystyi muuttamaan tekstin ääneksi. Laite kykeni parhaimmillaan prosessoimaan 60 sanaa minuutissa. [7]

Ensimmäisen sukupolven kaupallistetut tekstintunnistusjärjestelmät julkaistiin 1960-luvun alussa. Ensimmäinen näistä oli IBM:n julkaisema IBM-1418. Muiden ensimmäisen sukupolven järjestelmien tapaan IBM-1418 pystyi prosessoimaan vain erikseen sille kehitetyllä fontilla kirjoitettuja dokumentteja. [8]

Toisen sukupolven tekstintunnistusjärjestelmiä julkaistiin markkinoille 1960-luvun loppupuolella. Tämän sukupolven järjestelmille oli tyypillistä, että ne pystyivät tunnistamaan myös joitakin käsinkirjoitettuja merkkejä, jotka olivat yleensä numeroita. Eräs toisen sukupolven järjestelmä oli postikeskuksissa käytetty kirjeiden lajittelujärjestelmä. Kyseinen järjestelmä lajitteli kirjeitä niihin kirjoitettujen postinumeroiden perusteella. Parhaimmillaan tämä järjestelmä toimi 92 – 93 % tarkkuudella. [8]

1990-luvulle tultaessa kaupalliset tekstintunnistusjärjestelmät mahdollistivat 5-10 sivun prosessoimisen minuutissa. Parhaat järjestelmät lupasivat ihannetilanteessa tekstintunnistustarkkuudeksi jopa 99,9 %, mutta tämä saattoi olla harhaanjohtavaa, koska käytännössä tarkkuudet vaihtelivat suuresti muun muassa dokumenttien laadun ja eri fonttien takia. Käsinkirjoitettujen dokumenttien prosessointiin kykenevät järjestelmät olivat vielä 1990-luvulla harvinaisia. [8]

Hewlett-Packardin laboratoriossa Bristolissa aloitettiin 1980-luvulla tutkimus, jonka tarkoituksena oli kehittää uusi tekstintunnistusjärjestelmä. Tämän tutkimuksen pohjalta syntyi Tesseract-järjestelmä, joka oli 1990-luvun alussa merkittävästi vastaavia kaupallisia versioita tarkempi. Tesseractin kehitys kuitenkin lopetettiin yli kymmeneksi vuodeksi ja vuonna 2005 Hewlett-Packard julkaisi järjestelmän avoimen lähdekoodin palveluna [9]. Vaikka Tesseract oli jo tässä vaiheessa jäänyt jälkeen parhaista kaupallisista järjestelmistä, pystyy se vieläkin joissakin tapauksissa tarjoamaan vaihtoehdon kaupallisille järjestelmille. Walker et al. tekemässä tutkimuksessa vuonna 2018 Tesseractin normalisoitu merkkivirhesuhde (N-CER) englannin kielisten kirjojen kohdalla oli 1,0 %, kun taas kaupallisella Google Cloud Vision -palvelulla vastaava luku oli 0,6 %.[9, 12]

Viimeisimpien vuosien aikana useat kaupalliset toimijat ovat julkaisseet pilvipalveluympäristöissä toimivia tekstintunnistusjärjestelmiä. Amazon julkaisi Amazon Rekognition -palvelun vuonna 2016 ja seuraavana vuonna Google julkaisi Google Cloud Vision -palvelun. [10, 11].

3. PILVIPALVELUPOHJAISET OCR-PALVELUT

Pilvipalveluiden käyttö on kasvanut merkittävästi 2000-luvun aikana. Vuonna 2018 pilvipalvelumarkkinoiden arvo oli noin 182 miljardia dollaria ja sen ennustetaan nousevan 331 miljardiin dollariin vuoteen 2022 mennessä [34]. Caroll et al. mukaan kustannussäästöt ovat suurin yksittäinen tekijä, jonka takia yritykset siirtyvät pilvipalvelupohjaisiin ratkaisuihin [35]. Pilvipalveluiden ansiosta yritykset säästävät muun muassa sovelluskehityksessä ja ylläpidossa.

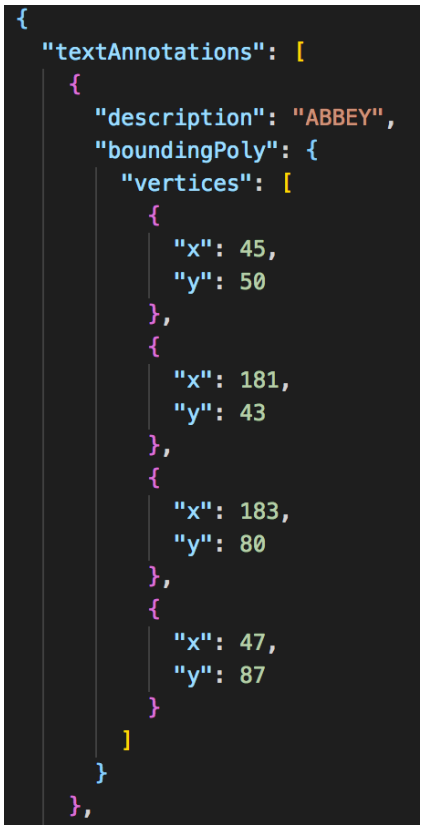
Pilvipalvelupohjaiset tekstintunnistusjärjestelmät mahdollistavat tekstintunnistuksen hyödyntämisen niin, ettei käyttäjän tarvitse itse ymmärtää, kuinka tekstintunnistus käytännössä toimii [13]. Han et al. julkaisivat helmikuussa 2019 tutkimuksen, jossa he tutkivat seitsemän eri tekstintunnistusjärjestelmän tarkkuutta, joista kolme oli pilvipalvelupohjaisia järjestelmiä [26]. Tutkittavia pilvipalvelujärjestelmiä olivat Abbyy Cloud, Google Cloud Vision ja Microsoft Azure Computer Vision. Han et al. mukaan kaikki palvelut suoriutuivat hyvälaatuisten dokumenttien tunnistamisesta hyvin [26]. Myös huonolaatuisten dokumenttien kohdalla useimmat palvelut pystyivät tunnistamaan dokumenttien sisältämän tekstin ymmärrettävästi, mutta tarkan tuloksen saavuttamiseksi täytyi käyttäjän tehdä jonkin verran korjauksia.

Tässä työssä vertaillaan kahta eri pilvipalvelupohjaista tekstintunnistusjärjestelmää. Vertailuun valitut palvelut ovat Google Cloud Vision ja Microsoft Azure Computer Vision.

3.1 Google Cloud Vision

Google Cloud Vision on vuonna 2017 julkaistu kaupallinen pilvipalvelupohjainen konenäköjärjestelmä, joka sisältää tekstintunnistuksen lisäksi muitakin palveluita [11]. Palvelu mahdollistaa dokumenttien analysoinnin REST-rajapinnan avulla. Rajapintakutsuissa ja -vastauksissa käytetään JSON-formaattia [27]. REST-rajapinnan lisäksi Google Cloud Vision tarjoaa Drag and Drop -palvelun, jolla voi analysoida yksittäisiä kuvia [25].

REST-rajapintaa käytettäessä kutsuun täytyy sisällyttää joko base64-koodattu kuva, Google Cloud Storage URI tai verkko-osoite, josta kuva on saatavilla. Vastaukseen rajapinta sisällyttää tiedon jokaisesta tunnistetusta sanasta ja sen sijainnista kuvassa. [28] Kuvassa 3 on esitetty eräs palvelun tunnistama sana ja sen sijainti alkuperäisessä dokumentissa.

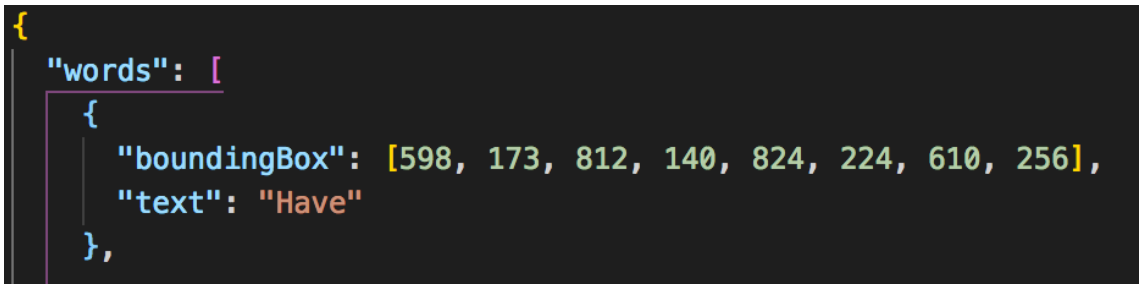


Kuva 3. *Osa Google Cloud Vision -palvelun REST-rajapinnan vastauksesta.*

Google Cloud Vision -palvelulla on mahdollista analysoida ilmaiseksi 1000 dokumenttia kuukaudessa. Tämän jälkeen analysointi maksaa 0,60 – 1,50 EUR / 1000 kpl.

3.2 Microsoft Azure Computer Vision

Microsoft Azure Computer Vision on vuonna 2016 julkaistu kuvien analysointi -palvelu, joka on osa Microsoftin Cognitive Services -palvelua [29]. Google Cloud Vision -palvelun tapaan myös Microsoft Azure Computer Vision -palvelua käytetään REST-rajapinnan avulla [32]. Palveluun on julkaistu uusi Read-API, jolla olisi mahdollista analysoida dokumentteja, mutta tämä ei sisällä vielä tukea suomenkielelle. Tästä syystä soveltuvuusvertailussa käytetään vanhempaa OCR-API:a, josta suomen kielen tuki löytyy. Kuvassa 4 on esitetty eräs palvelun OCR-API:n tunnistama sana ja sen sijainti alkuperäisessä dokumentissa.



Kuva 4. *Osa Microsoft Azure Computer Vision -palvelun REST-rajapinnan vastauksesta.*

Microsoft Azure Computer Vision -palvelulla on mahdollista analysoida ilmaiseksi 5000 dokumenttia kuukaudessa. Tämän jälkeen analysointi maksaa 0,549 – 1,265 EUR / 1000 kpl. [33]

4. SOVELTUVUUSVERTAILU

Soveltuvuusvertailu toteutetaan määrällisenä eli kvantitatiivisena tutkimuksena. Soveltuvuusvertailun tarkoituksena on selvittää, soveltuvatko vertailtavat pilvipalvelupohjaiset tekstintunnistusjärjestelmät mobiililaitteilla kuvattujen tositteiden käsittelyyn.

4.1 Aineisto

Soveltuvuusvertailussa käytettävä aineisto koostuu mobiililaitteilla kuvatuista tositteiden kuvista. Aineisto kerätään eTasku Solutions Oy:n hallinnoimasta tositearkistosta. Jotta kuvatut tositteet vastaisivat laadultaan nykyaikaisilla mobiililaitteilla kuvattuja kuvia, valitaan vertailun perusjoukoksi vuosien 2018 ja 2019 aikana kuvatut tositteet. Lisäksi aineiston yhtenäistämiseksi perusjoukko rajataan koskemaan käteiskaupan maksusuorituksista laadittuja tositteita eli kuitteja. Perusjoukon koko on tällöin noin 2,5 miljoonaa tositetta. Perusjoukosta valitaan tutkimusaineistoksi noin 100 tositetta yksinkertaisella satunnaisotantamenetelmällä.

4.2 Vertailussa käytettävät arviointikriteerit

Soveltuvuusvertailua varten määritellään tärkeimmät tositteisiin liittyvät tiedot. Laki kuitintarjoamisvelvollisuudesta käteiskaupassa määrittää, että käteiskaupan maksusuorituksesta laaditun kuitin täytyy sisältää seuraavat tiedot [23]:

1. elinkeinoharjoittajan nimi, yhteystiedot ja y-tunnus
2. kuitin antamispäivä
3. kuitin tunnistenumero tai muu yksilöivä tieto
4. myytyjen tavaroiden määrä ja laji sekä palvelujen laji
5. tavaroista tai palveluista suoritettu maksu ja suoritettavan arvonlisäveron määrä verokannoittain taikka arvonlisäveron peruste verokannoittain.

Soveltuvuusvertailussa keskitytään kaikkiin kohtien 1, 2 ja 5 määrittelemiin tietoihin. Yksittäistä kuitin sisältämää tietoa kutsutaan tässä työssä jatkossa tietoalkioksi. Elinkeinoharjoittajan yhteystiedoiksi hyväksytään puhelinnumero, katuosoite tai sähköposti. Soveltuvuusvertailussa tutkitaan, kuinka hyvin vertailtavat palvelut onnistuvat tunnistamaan tietoalkioita kuudessa eri tietoalkiokategoriassa: Kuitin kokonaishinta, kuitin antamispäivä, kuitin verokannan arvo sekä elinkeinoharjoittajan nimi, yhteystiedot ja y-tunnus.

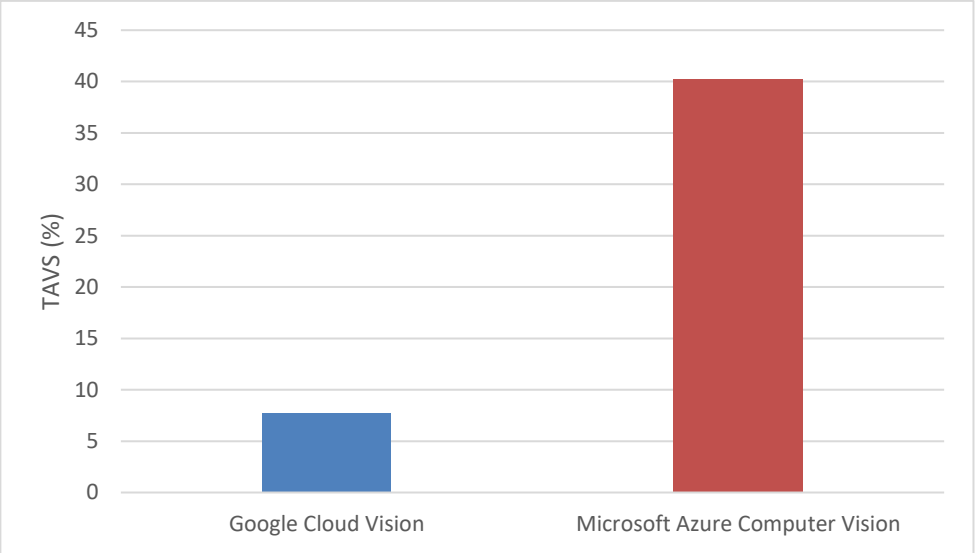
Arviointia varten määritellään tietoalkiovirhesuhde (TAVS), joka muodostetaan kaavan 3 avulla

$$TAVS = \frac{VTTA}{TAKM} * 100\ \%\ , \qquad (3)$$

jossa VTTA on virheellisesti tunnistettujen tietoalkioiden määrä ja TAKM on tietoalkioiden kokonaismäärä.

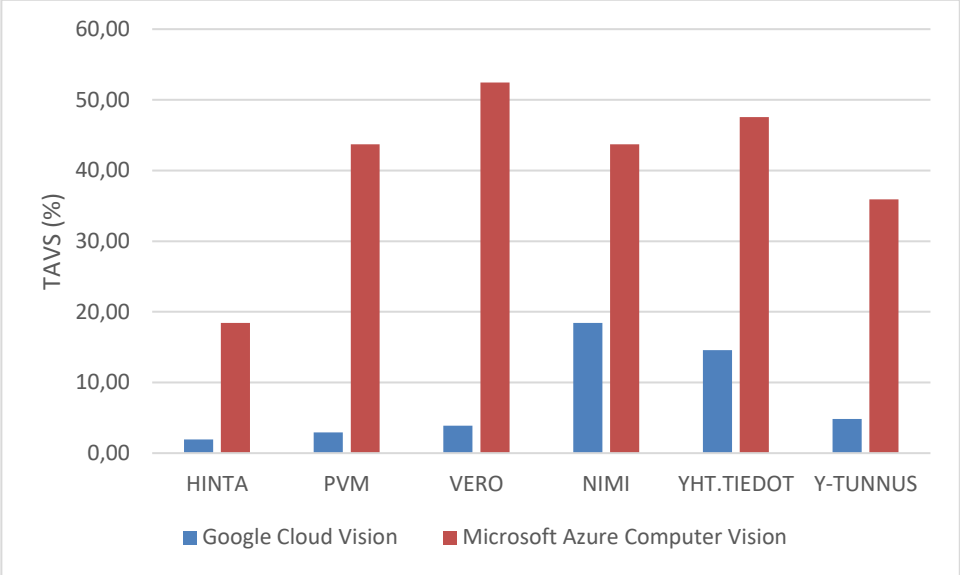
4.3 Tulokset

Kuvassa 5 on esitetty vertailtujen pilvipalvelupohjaisten tekstintunnistusjärjestelmien keskimääräiset tietoalkiovirhesuhteet. Google Cloud Vision -palvelulla keskimääräinen tietoalkiovirhesuhde oli 7,77 % ja Microsoft Azure Computer Vision -palvelulla vastaava luku oli 40,29 %. Keskimääräinen tietoalkiovirhesuhde on muodostettu laskemalla keskiarvo kaikista kuudesta eri tietoalkiokategorioiden virhesuhteista.



Kuva 5. *Palveluiden keskimääräiset tietoalkiovirhesuhteet.*

Kuvassa 4 on esitetty vertailtujen pilvipalvelujen tietoalkiovirhesuhteet eri vertailukategorioissa. Pienin virhesuhde oli Google Cloud Vision -palvelulla hintakategoriassa, jossa virhesuhde oli 1,94 %. Suurin virhesuhde Google Cloud Vision -palvelulla oli nimikategoriassa, jossa virhesuhde oli 18,45 %. Microsoft Azure Computer Vision -palvelun pienin virhesuhde oli myös hintakategoriassa, jossa virhesuhde oli 18,45 %. Suurin virhesuhde Microsoft Azure Computer Vision -palvelulla oli verokategoriassa, jossa virhesuhde oli 52,43 %.



Kuva 6. *Palveluiden tietoalkiovirhesuhteet eri kategorioissa.*

Taulukkoon 1 on koottu kaikki kuvien 5 ja 6 esittämät tietoalkiovirhesuhteet.

Taulukko 1. *Soveltuvuusvertailun tulokset.*

Palvelu	*Hinta*	*Pvm*	*Vero*	*Nimi*	*Yht.tie-dot*	*Y-tun-nus*	*Keskim.*
Google Cloud Vision	1,94	2,91	3,88	18,45	14,56	4,85	**7,77**
Microsoft Azure Compu-ter Vision	18,45	43,69	52,43	43,69	47,57	35,92	**40,29**

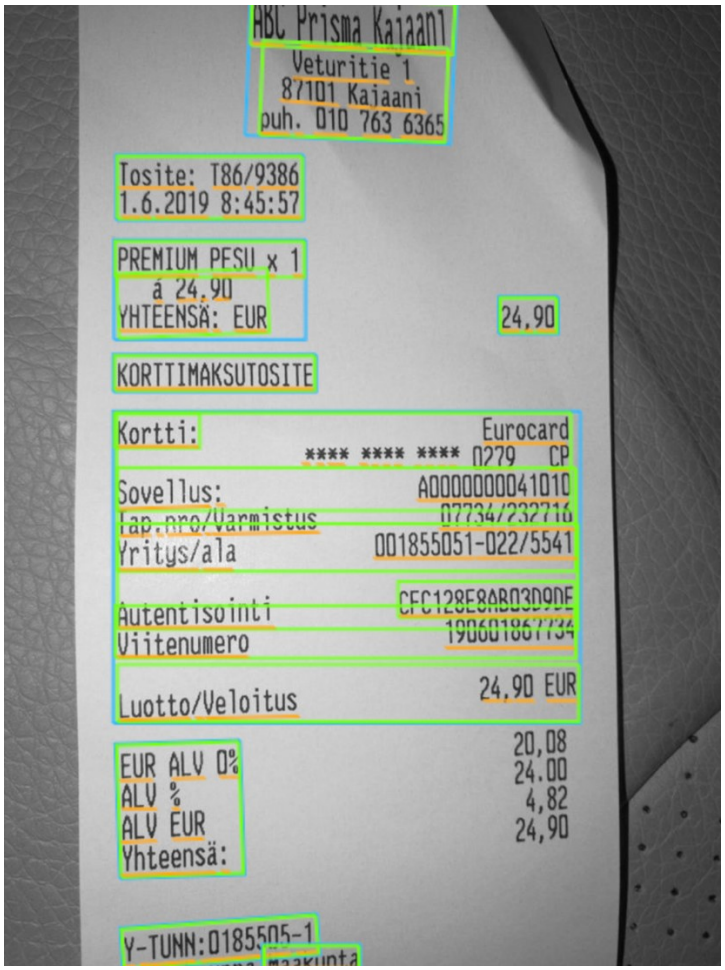
4.4 Tuloksien analysointi

Soveltuvuusvertailun tuloksien perusteella voidaan todeta, että Google Cloud Vision -palvelu suoriutui vertailusta huomattavasti Microsoft Azure Computer Vision -palvelua paremmin. Molemmissa palveluissa on huomattavissa, että numeroista koostuvan tekstin (hinta, päivämäärä, y-tunnus, verokanta) tunnistaminen onnistui paremmin kuin kirjaimista koostuvan (elinkeinoharjoittajan nimi ja yhteystiedot). Google Cloud Vision -palvelussa virheet syntyivät usein virheellisesti tulkituista merkeistä, kun taas Microsoft Azure Computer Vision -palvelussa jäi usein kokonaisia alueita tunnistamatta.

Vaikka Google Cloud Vision -palvelu onnistui yleensä tunnistamaan kaikki alueet kuitista, nousi soveltuvuusvertailun aikana esiin muutamia tapauksia, joissa kaikkien alueiden tunnistaminen ei onnistunut. Kuvassa 7 on esimerkki kuitista, jonka tunnistaminen Google Cloud Vision -palvelulla epäonnistui lähes kokonaan. Ainoastaan elinkeinoharjoittajan nimen tunnistaminen on onnistunut, mutta tämäkin sisältää merkkivirheen, kun nimessä oleva ä-kirjain on tulkittu a-kirjaimeksi. Kuitissa merkit koostuvat pisteistä. Tämä voisi olla mahdollinen syy tunnistuksen epäonnistumiselle.

Kuva 7. *Esimerkki täysin epäonnistuneesta tunnistuksesta [25].*

Kuvassa 8 on esitetty kuitti, jonka tunnistaminen Google Cloud Vision -palvelulla on epäonnistunut osittain. Kuitin oikean alalaidan ALV-erittelyn summat ovat jääneet tunnistamatta. Tämä voisi mahdollisesti johtua kuitin kaarevuudesta ja vinoumasta.



Kuva 8. *Esimerkki osittain epäonnistuneesta tunnistamisesta [25].*

Hyvälaatuisista kuvista, joissa kuitti on suoristettu, Google Cloud Vision -palvelu onnistui tunnistamaan käytännössä kaikki numeroista koostuvat tiedot. Elinkeinoharjoittajan nimen tai yhteystietojen tunnistuksen epäonnistuminen ei ole merkittävää, jos y-tunnuksen tunnistaminen onnistuu. Elinkeinoharjoittajan nimi ja yhteystiedot on mahdollista hakea y-tunnuksen esimerkiksi Patentti- ja rekisterihallituksen julkaiseman avoimen rajapinnan kautta [36].

5. YHTEENVETO

Tekstintunnistus koostuu useista monimutkaisista osa-alueista, joiden hallinta vaatii merkittävästi opettelua. Pilvipalvelupohjaiset tekstintunnistusjärjestelmät piilottavat tämän monimutkaisuuden tarjoten yksinkertaiset rajapinnat, joiden kautta tekstintunnistukseen perehtymätönkin voi hyödyntää palveluita.

Tämän työn soveltuvuusvertailun pohjalta voidaan todeta, että pilvipalvelupohjaisten tekstintunnistusjärjestelmien hyödyntäminen tositteiden käsittelyssä on mahdollista ainakin Google Cloud Vision -palvelun avulla. Microsoft Azure Computer Vision -palvelulla tositteiden käsittely ei ole kannattavaa. Vaikka Google Cloud Vision -palvelu suoriutui soveltuvuusvertailusta hyvin, jättää 7,77 % keskimääräinen tietoalkiovirhesuhde kehitettävää tulevaisuuteen.

Tulevaisuudessa tämän työn soveltuvuusvertailua voitaisiin kehittää muun muassa ottamalla huomioon, kuinka hyvin tositteissa toisiinsa liittyvät tiedot saataisiin yhdistettyä tekstintunnistusprosessin tuloksena saadusta datasta. Esimerkiksi tällaisia tietoja ovat verokantaotsikko ja kuitin verokannan arvo. Toinen mahdollinen tapa kehittää soveltuvuusvertailua olisi vertailla useampia pilvipalvelupohjaisia tekstintunnistusjärjestelmiä. Tätä työtä varten oli tavoitteena vertailla myös Amazon Textractia, josta on tällä hetkellä julkaistu rajattu testiversio. Testaamista varten olisi täytynyt päästä testiryhmään, mutta tämä ei kuitenkaan onnistunut ajoissa. Toinen mahdollinen testattava palvelu voisi olla Microsoft Azure Computer Vision -palvelun Read API, joka on uudempi kuin tässä työssä testattu OCR API, mutta se ei vielä sisällä tukea suomenkielelle.

LÄHTEET

[1] Sharma, S., Singh, N., Optical Character Recognition Using Artificial Neural Networks Approach, International Journal of Emerging Technology and Advanced Engineering, Volume 4, Issue 11, 2014. Saatavilla (viitattu 22.4.2019): https://ijetae.com/files/Volume4Issue11/IJETAE_1114_53.pdf

[2] Islam, N., Islam, Z., Noor, N., A Survey on Optical Character Recognition System, Journal of Information & Communication Technology-JICT Vol. 10 Issue. 2, 2016. Saatavilla (viitattu 22.4.2019): https://arxiv.org/pdf/1710.05703.pdf

[3] Bunke, H., Wang, P.S.P, Handbook Of Character Recognition And Document Image Analysis, 1997, pp. 1–47. Saatavilla (viitattu 22.4.2019): https://books.google.fi/books?hl=en&lr=&id=yn6DN5hAPywC&oi=fnd&pg=PA1&dq=Bunke,+H.,+Wang,+P.+S.+P.+(Editors),+Handbook+of+Character+Recognition+and+Document+Image+Analysis,+World+Scientific,+1997.&ots=QGftEiJDY&sig=JRdaXGZggfYPIfdY96CsZ8mO2BM&redir_esc=y#v=onepage&q=prepro&f=false

[4] Lund, W.B., Kennard, D.J., Ringer, E.K., Combining Multiple Thresholding Binarization Values to Improve OCR Output, Brigham Young University, Computer Science Department, Provo, Utah, USA, 2013. Saatavilla (viitattu 22.4.2019): https://pdfs.semanticscholar.org/ef3d/25693410a4e7c5261b670c33e9ef2e3fa00b.pdf

[5] Mokhtar, K., Bukhari, S.S., Dengel, A., OCR Error Correction: State-of-the-art vs An NMT Based Approach, 2018 13th IAPR International Workshop on Document Analysis Systems (DAS), 2018. Saatavilla (viitattu 1.5.2019): http://www.dfki.de/~bukhari/data/papers/71_DAS2018_OCR_Error.pdf

[6] Harding, S.M., Croft, W.B., Weir, C., Probabilistic Retrieval of OCR Degraded Text Using N-Grams, CIIR, University of Massachusetts, USA, 1997. Saatavila (viitattu 5.5.2019): https://s3.amazonaws.com/academia.edu.documents/30740569/Probabilistic_Retrieval_of_OCR_Degraded_Text.pdf?AWSAccessKeyId=AKIAIWOWYYGZ2Y53UL3A&Expires=1557050865&Signature=zO5tWWG7%2BxgIM9URZWIADB566PI%3D&response-content-disposition=inline%3B%20filename%3DProbabilistic_retrieval_of_ocr_degraded.pdf

[7] Fish, R.M., An Audio Display for the Blind, IEEE Transactions on Biomedical Engineering, Volume: BME-23, Issue: 2, 1976, pp. 144-154. Saatavilla (viitattu 5.5.2019): https://ieeexplore-ieee-org.libproxy.tuni.fi/stamp/stamp.jsp?tp=&arnumber=4121021

[8] Mori, S., Suen, C.Y., Yamamoto, K., Historical review of OCR research and development, Proceedings of the IEEE, Volume: 80, Issue: 7, 1992. Saatavilla (viitattu 5.5.2019): https://ieeexplore-ieee-org.libproxy.tuni.fi/stamp/stamp.jsp?tp=&arnumber=156468&tag=1

[9] Smith, R., An Overview of the Tesseract OCR Engine, Ninth International Conference on Document Analysis and Recognition (ICDAR 2007), 2007. Saatavilla (viitattu: 8.5.2019): https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=4376991

[10] Document History for Amazon Rekognition, Amazon Web Services, verkkosivu. Saatavilla (viitattu 8.5.2019): https://docs.aws.amazon.com/rekognition/latest/dg/document-history.html

[11] Cloud Vision Api Release Notes, Google Cloud, verkkosivu. Saatavilla (viitattu: 8.5.2019): https://cloud.google.com/vision/docs/release-notes

[12] Walker, J., Fujii Y., Popat A.C., A Web-Based OCR Service for Documents, 13th IAPR International Workshop on Document Analysis Systems, 2018. Saatavilla (viitattu 30.3.2019): https://das2018.cvl.tuwien.ac.at/media/filer_public/85/fd/85fd4698-040f-45f4-8fcc-56d66533b82d/das2018_short_papers.pdf#page=23

[13] What is Amazon Rekognition, Amazon Web Services, verkkosivu. Saatavilla (viitattu 8.5.2019): https://docs.aws.amazon.com/rekognition/latest/dg/what-is.html

[14] Ahmad, R., Faisal Rashid, S., Zeshan Afzal, M., Liwicki, M., Dengel, A., Breuel, T., A Novel Skew Detection and Correction Approach for Scanned Documents, DAS 2016, 12th Int'l IAPR Workshop on Document Analysis Systems, At Santorini, Greece, 2016. Saatavilla (viitattu 8.5.2019): https://www.researchgate.net/profile/Riaz_Ahmad9/publication/294578383_A_Novel_Skew_Detection_and_Correction_Approach_for_Scanned_Documents/links/5736081d08ae9f741b29cbd1/A-Novel-Skew-Detection-and-Correction-Approach-for-Scanned-Documents.pdf

[15] Jirasuwankul, N., Effect of Text Orientation to OCR Error and Anti-Skew of Text using Projective Transform Technique, 2011 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM2011), 2011. Saatavilla (viitattu 8.5.2019): https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=6027057

[16] Klink, S., Dengel, A., Kieninger, T., Document Structure Analysis Based on Layout and Textual Features, German Research Center for Artificial Intelligence, Kaiserslautern, Germany, 2000. Saatavilla (viitattu 8.5.2019): http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.71.1523&rep=rep1&type=pdf

[17] Mao, S., Kanungo, T., Empirical Performance Evaluation Methodology and Its Application to Page Segmentation Algorithms, IEEE Transactions on Pattern Analysis and Machine Intelligence (Volume: 23 , Issue: 3 , Mar 2001), 2001, pp. 242-256. Saatavilla (viitattu 19.5.2019): https://lhncbc.nlm.nih.gov/system/files/pub2001008.pdf

[18] Shafait, F., Keysers, D., Breuel, T.M., Performance Evaluation and Benchmarking of Six Page Segmentation Algorithms, IEEE Transactions on Pattern Analysis and Machine Intelligence (Volume: 30 , Issue: 6 , June 2008), 2008, pp. 941-954. Saatavilla (viirattu 19.5.2019): https://www.researchgate.net/profile/Thomas_Breuel/publication/5431819_Performance_Evaluation_and_Benchmarking_of_Six-Page_Segmentation_Algorithms/links/551dd6dd0cf213ef063eb1ee.pdf

[19] Nagy, G., Seth, S.C., Viswanathan, M., A Prototype Document Image Analysis System for Technical Journals, Computer (Volume: 25 , Issue: 7 , July 1992), 1992, pp. 10-22. Saatavilla (viitattu 19.5.2019): https://digitalcommons.unl.edu/cgi/viewcontent.cgi?article=1039&context=csearticles

[20] Kumar, G., Bhatia, P.K., A Detailed Review of Feature Extraction in Image Processing Systems, 2014 Fourth International Conference on Advanced Computing & Communication Technologies, India, 2014. Saatavilla (viitattu 19.5.2019): https://ieeexplore.ieee.org/abstract/document/6783417

[21] Arica, N., Yarman-Vural, F.T., An Overview of Character Recognition Focused on Off-Line Handwriting, IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews), 05/2001, Volume 31, Issue 2, 2001, pp. 216-233. Saatavilla (viitattu 19.5.2019): https://ieeexplore-ieee-org.libproxy.tuni.fi/stamp/stamp.jsp?tp=&arnumber=941845

[22] Cheriet, M., Kharma, N., Liu, C-L., Suen, C.Y., Character Recognition Systems: A Guide for Students and Practitioners, New Jersey, USA, 2007. Saatavilla (viitattu 19.5.2019): https://pdfs.semanticscholar.org/74d6/68256131f379d63a3d484ccff513f5bbb6d3.pdf

[23] Laki kuitintarjoamisvelvollisuudesta käteiskaupassa 2013/658 § 4. Saatavilla (viitattu 30.5.2019): https://www.finlex.fi/fi/laki/ajantasa/2013/20130658

[24] Klein, B., Levenshtein Distance, Python Advanced Course Topics, verkkosivu. Saatavilla (viitattu 30.5.2019): https://www.python-course.eu/levenshtein_distance.php

[25] Drag and drop, Google Cloud Vision API, verkkosivu. Saatavilla (viitattu 7.6.2019): https://cloud.google.com/vision/docs/drag-and-drop

[26] Han, T., Hickman, A., Our search for the best OCR tool, and What We Found, Source, verkkosivu. Saatavilla (viitattu 8.6.2019): https://source.opennews.org/articles/so-many-ocr-options/

[27] Make a Vision API request, Google Cloud, verkkosivu. Saatavilla (viitattu 8.6.2019): https://cloud.google.com/vision/docs/request

[28] Detect Text (OCR), Google Cloud, verkkosivu. Saatavilla (viitattu 8.6.2019): https://cloud.google.com/vision/docs/ocr

[29] Public preview: Computer Vision API and Academic Knowledge API in Cognitive Services, Microsoft, verkkosivu. Saatavilla (viitattu 8.6.2019): https://azure.microsoft.com/en-us/updates/public-preview-microsoft-cognitive-services-computer-vision-api-and-academic-knowledge-api/

[30] Verma, R., Ali, D.J., A-survey of feature extraction and classification techniques in OCR systems. International Journal of Computer Applications & Information Technology, 2012. Saatavilla (viitattu 9.6.2019): http://www.ijcait.com/IJCAIT/13/131.pdf

[31] Yetirajam, M., Nayak, M. R., Chattopadhyay, S., Recognition and classification of broken characters using feed forward neural network to enhance an OCR solution. International Journal of Advanced Research in Computer Engineering & Technology (IJARCET) Volume, 1, 2012. Saatavilla (viitattu 9.6.2019): https://pdfs.semanticscholar.org/74ee/3b46b4a2ee58426cda1715d71f5b9d621f97.pdf

[32] What is Computer Vision, Micorosft, verkkosivu. Saatavilla (viitattu 9.6.2019): https://docs.microsoft.com/en-in/azure/cognitive-services/computer-vision/home

[33] Cognitive Services Pricing-Computer Vision API, Microsoft, verkkosivu. Saatavilla (viitattu: 9.6.2019): https://azure.microsoft.com/en-in/pricing/details/cognitive-services/computer-vision/

[34] Columbus, L., Public Cloud Soaring To $331B By 2022 According To Gartner, Forbes, 2019, verkkosivu. Saatavilla (viitattu 20.6.2019): https://www.forbes.com/sites/louiscolumbus/2019/04/07/public-cloud-soaring-to-331b-by-2022-according-to-gartner/

[35] Carroll, M., Van Der Merwe, A., Kotze, P, Secure cloud computing: Benefits, risks and controls. In 2011 Information Security for South Africa (pp. 1-9). IEEE. 2011. Saatavilla (viitattu 20.6.2019): https://researchspace.csir.co.za/dspace/bitstream/handle/10204/5184/Kotze4_2011.pdf?sequence=1

[36] Tietoa avoimen datan rajapinnoista, Patentti- ja rekisterihallitus, verkkosivu. Saatavilla (viitattu: 20.6.2019): http://avoindata.prh.fi/index.html